Amendment and Response
Applicant: Norman C. Chou et al.
Serial No.: 10/057,159
Filed: January 24, 2002
Docket No.: 10011314-1/A310.258.101
Title: CONTROL OF AUTHENTICATION DATA RESIDING IN A NETWORK DEVICE

IN THE SPECIFICATION

Please replace the paragraph beginning at paragraph [0003] with the following rewritten paragraph:

[0003] With a view to meeting the above described challenges, a new interconnect technology, called the InfiniBand™, has been proposed for interconnecting processing nodes and I/O nodes to form a System Area Network (SAN). This architecture has been designed to be independent of a host Operating System (OS) and processor platform. The InfiniBand™ Architecture (IBA) is centered around a point-to-point, switched IP fabric whereby end node devices (e.g., inexpensive I/O devices such as a single chip Small Computer System Interface (SCSI) or Ethernet adapter, or a complex computer system) may be interconnected utilizing a cascade of switch devices. The InfiniBand™ Architecture is defined in the InfiniBand™ Architecture Specification Volume 1, Release 1.0, released October 24, 2000 by the InfiniBand Trade Association. The IBA supports a range of applications ranging from back plane interconnect of a single host, to complex system area networks, as illustrated in **Figure 1** (prior art). In a single host environment, each IBA switched fabric may serve as a private I/O interconnect for the host providing connectivity between a CPU and a number of I/O modules. When deployed to support a complex system area network, multiple IBA switch fabrics may be utilized to interconnect numerous hosts and various I/O units.

Please replace the paragraph beginning at paragraph [0004] with the following rewritten paragraph:

[0004] Within a switch fabric supporting a System Area Network as indicated in 10, such as that shown in **Figure 1**, there may be a number of devices having multiple input and output ports through which data (e.g., packets) is directed from a source device to a destination device. Such devices include, for example, switches 130, routers, repeaters and adapters 110 and 120 (exemplary interconnect devices). In addition to multiple communication ports directing external data packets, an interconnect device such as a switch typically includes a management port. Each sub-network (subnet) is managed by at least one Subnet Manager. A Subnet Manager resides either on an endnode or on an interconnect device and can be implemented

either in hardware or software. The Subnet Manager performs its managing functions by communicating with the management port using InfiniBand™ Subnet Management Packets.

Please replace the paragraph beginning at paragraph [0018] with the following rewritten paragraph:

[0018] Note also that embodiments of the present description may be implemented no only within a physical circuit (e.g., on semiconductor chip) but also within machine-readable media. For example, the circuits and designs discussed above may be stored upon and/or embedded within machine-readable media associated with a design tool used for designing semiconductor devices. Examples include a netlist formatted in the Very High Speed Integrated Circuit (VHSIC), Hardware Description Language (VHDL) language, Verilog language or SPICE language. Some netlist examples include: a behavioral level netlist, a register transfer level (RTL) netlist, a gate level netlist and a transistor level netlist. Machine-readable media also include media having layout information such as a GDS-II file. Furthermore, netlist files or other machine-readable media for semiconductor chip design may be used in a simulation environment to perform the methods of the teachings described above.

Please replace the paragraph beginning at paragraph [0024] with the following rewritten paragraph:

[0024] The present invention provides a mechanism that allows subnet manager 206 or any other authorized entity to regain control over interconnect device 202 at any time. Specifically, a configuration switch 204 is provided which is responsible for receiving an operator's command to reset the authentication data stored in interconnect device 2020 and generating a ~~rest~~reset signal in response to the operator's command. Interconnect device 202 receives the reset signal from configuration switch 204 and resets the authentication data, thereby returning interconnect device 202 into a state that allows subnet manager 206 or any other authorized entity to establish new authentication data within interconnect device 202. In one embodiment, the configuration switch 204 includes a pin located outside of interconnect device 202. A change of the pin's state (i.e., the pin is being either set or cleared) causes a generation of

a reset signal.

Please replace the paragraph beginning at paragraph [0028] with the following rewritten paragraph:

[0028] Yet other agents residing in management port 302 include a decoder 304, ~~and a~~ subnet management agent (SMA) 308 and agent 346. Decoder 304 is responsible for decoding and dispatching data packets received at management port 302 to destination agents within management port 302. SMA 308, and agent 346 are ~~is a~~ targeting destination agents for subnet management packets (SMPs) sent by a subnet manager 332. Each of decoder 304, ~~and~~ SMA 308, and agent 346 includes a copy of the management key and associated attributes. When decoder 304 determines that a data packet being decoded is a SMP, decoder 304 compares its copy of the management key with the management key included in the packet. If the two management keys match, decoder 304 forwards the SMP to SMA 308. Otherwise, decoder 304 discards the SMP. This authentication check is not performed when the decoder's copy of the management key is set to zero.

Please replace the paragraph beginning at paragraph [0033] with the following rewritten paragraph:

[0033] **Figure 4** is an exemplary datagram 400 of a SMP received by decoder 302 of **Figure 3**, according to one embodiment of the present invention.

Please replace the paragraph beginning at paragraph [0034] with the following rewritten paragraph:

[0034] Referring to **Figure 4**, the positions of each field within the packets is provided in bits words. When there are two numbers, the number in parenthesis is given for a packet without a global router header (GRH), and the other number is given for a packet that ~~does not~~ includes a GRH.

Please replace the paragraph beginning at paragraph [0039] with the following rewritten paragraph:

[0039] Method 500 beings at block 502 with the processing logic receiving a reset signal from a configuration switch coupled to the management port (processing block 504). The reset signal indicates that an operator has requested to reset authentication data stored in a decoder of the management port. As described in more detail above, the authentication data facilitates the authorization of management operations associated with the interconnect device. In one embodiment, the authentication data is represented as a management key.

Please replace the paragraph beginning at paragraph [0042] with the following rewritten paragraph:

[0042] Afterwards, at processing block 510, the processing logic sets the decoder's copy of the management key to the new value. In one embodiment, the update of the decoder's copy of the management key is performed upon receiving an update request from an initialization agent of the management key. Specifically, once the decoder receives a valid data packet from the subnet manager, it decodes the data packet and sends the data packet to a subnet management agent (SMA) residing in the management port. The SMA determines that the data packet includes a request to update the authentication data, updates its copy of the authentication data, and notifies the initialization agent about this request. The initialization agent updates its own authentication data and sends a command to update a corresponding copy of the authentication data to each unit of the interconnect device that stores such a copy, including the decoder. As a result, the mismatch between the authentication data maintained by the subnet manager and the authentication data maintained by the interconnect device is corrected, and normal authentication operations can resume. The method ends at block 512.

Please replace the paragraph beginning at paragraph [0044] with the following rewritten paragraph:

[0044] Method 600 begins at block 602 with the processing logic detecting that a reset of authentication data (e.g., a management key) maintained by a management port of the

interconnect device is required (processing block 604). In one embodiment, the reset requirement is detected upon receiving a trap indicating that the management port has invalidated an initial data packet (e.g., a SMP) sent to the management port by the subnet manager due to a violation of the authentication data. In one embodiment, the trap is issued when the management port detects a mismatch between the authentication data stored in the management port and the authentication data included in the initial data packet, and an expiration attribute associated with the authentication data is set to the value (e.g., a zero) that prevents the expiration of the authentication data as explained in greater detail above.

Please replace the paragraph beginning at paragraph [0048] with the following rewritten paragraph:

[0048] Once the processing logic determines that the reset is required, it prevents the transmission of data packets to the management port (processing logic 608) until receiving a message from the operator that indicates that the authentication data maintained by the management port has been reset (processing block 610). Subsequent to the operators' message, the processing logic sends to the management port an update data packet with a request to set the authentication data maintained by the interconnect device to an update value (processing block 612). In one embodiment, the update value is the value stored in a database of the subnet manager. In another embodiment, prior to sending the update data packet, the processing logic determines the update value. In this embodiment, the operator's reset command causes only the reset of the decoder's copy of the authentication data; other agents of the management port still store valid authentication data, as described in greater detail above. In this embodiment, once the processing logic receives the operator's message indicating that the reset has been performed, the processing logic sends to the management port a read data packet requesting the current value of the authentication data maintained by the management port. In one embodiment, this data packet is processed by the SMA of the management port that queries the initialization agent for the current value of the authentication data and sends a response with this value back to the subnet manager. The processing logic then updates the database of the subnet manager with the received value of the authentication data and also uses this value as an update value in the update

data packet that is sent to the management port at processing block 612. The method ends at block 614.

Please replace the paragraph beginning at paragraph [0049] with the following rewritten paragraph:

[0049] **Figures 7A and 7B** illustrate the operation of some embodiments of the present invention using two exemplary scenarios shown at 700 and 720.

Please replace the paragraph beginning at paragraph [0052] with the following rewritten paragraph:

[0052] Master subnet manager 706 maintains a database 710 where the management key associated with interconnect device 702 is stored. When master subnet manager 706 inadvertently goes away, a transition to a backup subnet manager 708 takes place. During this transition, the correct management key may be lost (e.g., database 710 may be lost or its data may be contaminated), leaving backup subnet manager 708 with an incorrect management key. Subsequently, when backup subnet manager 708 sends an SMP to the SMA of the management port, the authentication check performed by the decoder will fail. If at this time, the expiration attribute is set to zero at this time, backup subnet manager 708 will detect that the management key is violated using one of the mechanisms described above and will notify the operator that the reset of the management key residing in the management port is needed. The operator will then send a reset command to the management port via a configuration switch ~~204~~704.

Please replace the paragraph beginning at paragraph [0054] with the following rewritten paragraph:

[0054] Referring to **Figure 7B**, subnet 730 includes interconnect device 734 with associated management key 732. ~~the~~ The size of subnet 730 is increased by adding an interconnect device 722 initially included in subnet 720. The management key associated with interconnect device 722 is known to a subnet manager 724 but not a subnet manager 736. In one embodiment, to avoid the mismatch of management keys, the installation procedure requires the

operator performing the addition to interconnect device ~~726~~ 722 to subnet 730 to reset the management key 726 using a configuration switch 728. In another embodiment, the installation procedure does not require the reset of the management key. Instead, subnet manager 736 detects that such reset is required after the installation of interconnect device 726 is completed and notifies the operator about the reset requirement as described in greater detail above.